

# PyWPS 4

## Development restart

Jáchym Čepický<sup>1</sup>

<sup>1</sup><http://les-ejk.cz>

Geoinformatics, Prague 2013



## 1 PyWPS

## 2 PyWPS 4



- OGC WPS on the Server
- Since 2006
- Python
- <http://pywps.wald.intevation.org>
- <http://github.org/geopython/pywps>

PyWPS

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated

PyWPS

# PyWPS - what it is NOT

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated

PyWPS

# PyWPS - what it is NOT

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated

PyWPS

PyWPS

# Keywords



rather bike, then a car

PyWPS





#small

PyWPS

# Keywords



#modular

PyWPS



#fast

PyWPS



#easy

PyWPS



#slick

PyWPS

## Keywords



#accessories (GRASS, GDAL, Shapely, #python) PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

PyWPS



#geopython 2006

PyWPS



#geopython 2013

PyWPS

- lxml <http://lxml.org>
- GRASS-WPS, GRASS-Python
- Werkzeug <http://werkzeug.pocoo.org/>
- Python 3
- Django
- ...

PyWPS



# Process definition in PyWPS 3

```
class Buffer(WPSProcess):  
    def __init__(self):  
        WPSProcess.__init__(self, identifier="buffer")  
        self.addComplexInput(identifier="input", ..  
        ...  
        self.addLiteralOutput(identifier="output", ..  
        ...  
    def execute(self):  
        ...  
    return
```

PyWPS

# Process definition in PyWPS 4 (proposal)

- 2-level APIs
- Providing the low-level API (integration with GRASS, writing a compatibility api for PyWPS-3, and defining a bunch of processes programatically if the user needs that.
- The process is just a callable
- Inspired by Django
- Upper-level API for easy user-written processes

PyWPS

```
class Buffer(pywps.Process):  
    class Input:  
        distance = pywps.Literal(value_type=pywps.F  
        layer = pywps.Complex(as_file=True, max_occ  
    class Output:  
        feature_count = pywps.Literal(value_type=py  
        fortune = pywps.Literal(value_type  
    def execute(self):  
        ...
```

PyWPS

```
def make_buffer(request):  
    ...  
make_buffer_process = pywps.create_process(  
    identifier='makebuffer', title=...  
    inputs=[  
        pywps.Literal(identifier='distance', value_  
        pywps.BoundingBox(identifier='clip'),  
    ],  
    outputs=[  
        pywps.Literal(identifier='feature_count', v  
        pywps.Literal(identifier='fortune', value_ty  
    ],  
    handler=make_buffer)  
make_buffer_process()
```

PyWPS

# Where, what, how, who?

- <http://github.org/jachym/pywps-4>
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- <http://lists.osgeo.org/cgi-bin/mailman/listinfo/pywps-dev>

PyWPS

# Where, what, how, who?

- <http://github.org/jachym/pywps-4>
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- <http://lists.osgeo.org/cgi-bin/mailman/listinfo/pywps-dev>

PyWPS

# Where, what, how, who?

- <http://github.org/jachym/pywps-4>
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- <http://lists.osgeo.org/cgi-bin/mailman/listinfo/pywps-dev>

PyWPS

# Where, what, how, who?

- <http://github.org/jachym/pywps-4>
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- <http://lists.osgeo.org/cgi-bin/mailman/listinfo/pywps-dev>

PyWPS



thank you!  
jachym@les-ejk.cz  
<http://github.org/jachym/pywps-4>

questions?

PyWPS